EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	763	717/130-131.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:13
L2	8	717/130-131.ccls. and (data adj1 race)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:13
L8	13	717/130-131.ccls. and (race and((multi$3 adj1 thread) concurren$4))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:21
L9	17	717/124.ccls. and (race and((multi$3 adj1 thread) concurren$4))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:22
L10	3	717/136.ccls. and (race and((multi$3 adj1 thread) concurren$4))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:22
L11	7	717/139-140.ccls. and (race and((multi$3 adj1 thread) concurren$4))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:22
L12	38	714/36-39.ccls. and (race and((multi$3 adj1 thread) concurren$4))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:23
L13	0	714/719.ccls. and (race and((multi$3 adj1 thread) concurren$4))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:23
L14	42	718/100.ccls. and (race and((multi$3 adj1 thread) concurren$4))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/24 20:23

EAST Search History

S1	51	("6405326" "6851075" "20050177775" "20030131283" "6817009" "20020129306" "20050216798" "6343371" "6393440" "5632034" "5918243" "5924098" "6289360" "6345351" "6665708" "6920541" "6928534" "20020120823" "20040172626" "20020120428" "20050283780" "20050283781" "20060200823" "20040123185" "20060161897" "7100164" "6779135" "6826752" "20040078785" "4855903" "6088511" "6106575" "4901224" "5197137" "5222235" "5247689" "5581458" "6047356" "4399507" "4446518" "4456958" "4466061" "4468736" "4498136" "4773041" "4783734" "4873629" "4975690" "5179672" "5222221").pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/19 16:25
S2	20	S1 and (race and ((multi adj5 thread)concurrent$3))	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/19 16:27
S3	206	(race with detect$4) and ((multi adj5 thread$3)concurrent$4) and sequent$5	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/19 17:27
S4	1	10/765717	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/22 16:16
S5	1	10/695970	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/23 14:29
S6	0	"data adj1 race" and ((multi$5 adj1 thread) concurren$3)and sequent$5 and stack and (pointer with thread)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/23 14:34
S7	7	(data adj1 race) and ((multi$5 adj1 thread) concurren$3)and sequent$5 and stack and (pointer with thread)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/01/23 14:34

Terms used **Multi Thread concurrent** and **data race**

Found **19,063** of **195,947**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Testing races in parallel programs with an OtOt strategy

Suresh K. Damodaran-Kamal, Joan M. Francioni

August 1994 **Proceedings of the 1994 ACM SIGSOFT international symposium on Software testing and analysis ISSTA '94**

Publisher: ACM Press

Full text available: pdf(1.13 MB) Additional Information: full citation, references, citings, index terms

2 Multi-thread graph: a system model for real-time embedded software synthesis

F. Thoen, J. Van Der Steen, G. de Jong, G. Goossens, H De Man

March 1997 **Proceedings of the 1997 European conference on Design and Test EDTC '97**

Publisher: IEEE Computer Society

Full text available: pdf(758.90 KB) Publisher Site Additional Information: full citation, abstract, citings

Software synthesis is a new approach which focuses on the support of real-time embedded multi-tasking software without the use of operating systems. A software synthesis system starts from a concurrent process system specification and maps this description automatically onto one or more processors. In this paper the internal system-level model which captures the embedded software and which is the backbone of our software synthesis methodology, is presented. The model captures the fine-grain beha ...

Keywords: concurrency, concurrent process system specification, control flow concepts, data communication, fine-grain behaviour, hierarchy constraints, internal system-level model, multi-tasking software, multi-thread graph, real-time embedded software synthesis, real-time systems, software synthesis methodology, synchronisation, system model, timing constraints

3 On-the-fly detection of data races for programs with nested fork-join parallelism

John Mellor-Crummey

August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing Supercomputing '91**

Publisher: ACM Press

Full text available: pdf(1.06 MB) Additional Information: full citation, references, citings, index terms

4 Ownership types for safe programming: preventing data races and deadlocks

Chandrasekhar Boyapati, Robert Lee, Martin Rinard

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '02**, Volume 37 Issue 11

Publisher: ACM Press

Full text available: pdf(459.57 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a new static type system for multithreaded programs; well-typed programs in our system are guaranteed to be free of data races and deadlocks. Our type system allows programmers to partition the locks into a fixed number of equivalence classes and specify a partial order among the equivalence classes. The type checker then statically verifies that whenever a thread holds more than one lock, the thread acquires the locks in the descending order.Our system also allows programmer ...

Keywords: data races, deadlocks, encapsulation, ownership types

5 RaceTrack: efficient detection of data race conditions via adaptive tracking

Yuan Yu, Tom Rodeheffer, Wei Chen

October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5

Publisher: ACM Press

Full text available: pdf(321.34 KB) Additional Information: full citation, abstract, references, citings, index terms

Bugs due to data races in multithreaded programs often exhibit non-deterministic symptoms and are notoriously difficult to find. This paper describes RaceTrack, a dynamic race detection tool that tracks the actions of a program and reports a warning whenever a suspicious pattern of activity has been observed. RaceTrack uses a novel hybrid detection algorithm and employs an adaptive approach that automatically directs more effort to areas that are more suspicious, thus providing more accurate war ...

Keywords: race detection, virtual machine instrumentation

6 Verification: Automated type-based analysis of data races and atomicity

Amit Sasturkar, Rahul Agarwal, Liqiang Wang, Scott D. Stoller

June 2005 **Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '05**

Publisher: ACM Press

Full text available: pdf(259.74 KB) Additional Information: full citation, abstract, references, citings, index terms

Concurrent programs are notorious for containing errors that are difficult to reproduce and diagnose at run-time. This motivated the development of type systems that statically ensure the absence of some common kinds of concurrent programming errors including data races and atomicity violations. A method is atomic if every execution of the concurrent program is equivalent to an execution in which the atomic method is executed without being interleaved with other concurrently executed methods. At ...

Keywords: atomicity, data races, type inference, type system

7 Race Frontier: reproducing data races in parallel-program debugging

Jong-Deok Choi, Sang Lyul Min

April 1991 **ACM SIGPLAN Notices , Proceedings of the third ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '91**, Volume 26 Issue 7

Publisher: ACM Press

Full text available: pdf(1.05 MB) Additional Information: full citation, references, citings, index terms

8 Eraser: a dynamic data race detector for multithreaded programs

Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro, Thomas Anderson

November 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 4

Publisher: ACM Press

Full text available: pdf(136.04 KB) Additional Information: full citation, abstract, references, citings, index terms

Multithreaded programming is difficult and error prone. It is easy to make a mistake in synchronization that produces a data race, yet it can be extremely hard to locate this mistake during debugging. This article describes a new tool, called Eraser, for dynamically detecting data races in lock-based multithreaded programs. Eraser uses binary rewriting techniques to monitor every shared-monory reference and verify that consistent locking behavior is observed. We present several case studies ...

Keywords: binary code modification, multithreaded programming, race detection

9 Eraser: a dynamic data race detector for multi-threaded programs

Stefan Savage, Michael Burrows, Greg Nelson, Patrick Sobalvarro, Thomas Anderson

October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles SOSP '97**, Volume 31 Issue 5

Publisher: ACM Press

Full text available: pdf(1.51 MB) Additional Information: full citation, references, citings, index terms

10 Doctoral symposium: Efficient data race and deadlock prevention in concurrent object-oriented programs

Piotr Nienaltowski

October 2004 **Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '04**

Publisher: ACM Press

Full text available: pdf(171.10 KB) Additional Information: full citation, abstract, references, index terms

The main goal of this PhD thesis is to propose and implement a methodology for the construction of programs based on the SCOOP model, and for modular reasoning about their correctness and liveness properties. In particular, the set of correctness rules that guarantee the absence of data races will be refined and formalized; an augmented type system will be proposed to enforce these rules at compile time. Furthermore, an efficient methodology for deadlock prevention, avoidance, detection, and ...

Keywords: Eiffel, SCOOP model, data races, deadlocks, object-oriented concurrency, ownership types

11 Concurrency analysis in the presence of procedures using a data-flow framework

Evelyn Duesterwald, Mary Lou Soffa

October 1991 **Proceedings of the symposium on Testing, analysis, and verification TAV4**

Publisher: ACM Press

Full text available: pdf(1.10 MB) Additional Information: full citation, references, citings, index terms

12 Efficient on-the-fly data race detection in multithreaded C++ programs

Eli Pozniansky, Assaf Schuster

June 2003 **ACM SIGPLAN Notices , Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '03,** Volume 38 Issue 10

Publisher: ACM Press

Full text available: pdf(288.53 KB) Additional Information: full citation, abstract, references, citings, index terms

Data race detection is highly essential for debugging multithreaded programs and assuring their correctness. Nevertheless, there is no single universal technique capable of handling the task efficiently, since the data race detection problem is computationally hard in the general case. Thus, to approximate the possible races in a program, all currently available tools take different ``short-cuts'', such as using strong assumptions on the program structure or applying various heuristics. When app ...

Keywords: concurrency, data race, instrumentation, multithreading, synchronization

13 Associating synchronization constraints with data in an object-oriented language

Mandana Vaziri, Frank Tip, Julian Dolby

January 2006 **ACM SIGPLAN Notices , Conference record of the 33rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '06,** Volume 41 Issue 1

Publisher: ACM Press

Full text available: pdf(254.75 KB) Additional Information: full citation, abstract, references, index terms

Concurrency-related bugs may happen when multiple threads access shared data and interleave in ways that do not correspond to any sequential execution. Their absence is not guaranteed by the traditional notion of "data race" freedom. We present a new definition of data races in terms of 11 problematic interleaving scenarios, and prove that it is *complete* by showing that any execution not exhibiting these scenarios is serializable for a chosen set of locations. Our definition subsumes the ...

Keywords: concurrent object-oriented programming, data races, programming model, serializability

14 Debugging concurrent programs

Charles E. McDowell, David P. Helmbold

December 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 4

Publisher: ACM Press

Full text available: pdf(2.86 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The main problems associated with debugging concurrent programs are increased complexity, the "probe effect," nonrepeatability, and the lack of a synchronized global clock. The probe effect refers to the fact that any attempt to observe the behavior of a distributed system may change the behavior of that system. For some parallel programs,

different executions with the same data will result in different results even without any attempt to observe the behavior. Even when the behavior can be ...

15 Concurrent real-time programming: Scheduling-independent threads and exceptions in SHIM

Olivier Tardieu, Stephen A. Edwards

October 2006 **Proceedings of the 6th ACM & IEEE International conference on Embedded software EMSOFT '06**

Publisher: ACM Press

Full text available: pdf(167.67 KB) Additional Information: full citation, abstract, references, index terms

Concurrent programming languages should be a good fit for embedded systems because they match the intrinsic parallelism of their architectures and environments. Unfortunately, typical concurrent programming formalisms are prone to races and nondeterminism, despite the presence of mechanisms such as monitors.In this paper, we propose SHIM, the core of a deterministic concurrent language, meaning the behavior of a program is independent of the scheduling of concurrent operations. SHIM does not sac ...

Keywords: deterministic model of computation, hardware/software codesign

16 Testing and debugging: Efficient event generation for detecting races

Scotte Zinn, Michael Coffin

October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1 CASCON '93**

Publisher: IBM Press

Full text available: pdf(746.03 KB) Additional Information: full citation, abstract, references

A *data race* is a situation during the execution of a parallel program where a variable is accessed concurrently by two independent threads and at least one of the accesses is a write operation. Races are a common cause of errors in parallel programs, so detecting them is an essential debugging task.A brute-force strategy used in many postmortem race detection methods involves instrumenting the program to record each inter-thread synchronization and each shared variable access in a log fil ...

17 Protocol-based data-race detection

Brad Richards, James R. Larus

August 1998 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools SPDT '98**

Publisher: ACM Press

Full text available: pdf(1.04 MB) Additional Information: full citation, references, citings, index terms

18 Detecting data races on weak memory systems

Sarita V. Adve, Mark D. Hill, Barton P. Miller, Robert H. B. Netzer

April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture ISCA '91**, Volume 19 Issue 3

Publisher: ACM Press

Full text available: pdf(1.15 MB) Additional Information: full citation, references, citings, index terms

19 Improving the accuracy of data race detection

Robert H. B. Netzer, Barton P. Miller

April 1991 **ACM SIGPLAN Notices , Proceedings of the third ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '91**, Volume 26 Issue 7

Publisher: ACM Press

Full text available: pdf(1.37 MB) Additional Information: full citation, references, citings, index terms

20 Grid and races: A theory of data race detection

Utpal Banerjee, Brian Bliss, Zhiqiang Ma, Paul Petersen

July 2006 **Proceeding of the 2006 workshop on Parallel and distributed systems: testing and debugging PADTAD '06**

Publisher: ACM Press

Full text available: pdf(263.11 KB) Additional Information: full citation, abstract, references, index terms

This paper presents a rigorous mathematical theory for the detection of data races in threaded programs. After creating a structure with precise definitions and theorems, it goes on to develop four algorithms with the goal of detecting at least one race in the situation where the history kept on previous memory accesses is limited. The algorithms demonstrate the tradeoff between the amount of access history kept and the kinds of data races that can be detected. One of these algorithms is a refor ...

Keywords: access conflict, data race, dependence, happens-before, synchronization, thread, vector clock